Section 3.9 Experiments with Quantitative Factors, Goodness of Fit

Yibi Huang

3.9 Experiments with Quantitative Factors, Goodness of Fit (Dose Response Modeling)

Example — Resin Glue Failure Time — Background

- How to measure the lifetime of things like computer disk drives, light bulbs, and glue bonds?
  E.g., a computer drive is claimed to have a lifetime of 800,000 hours ($>$ 90 years).
  Clearly the manufacturer did not have disks on test for 90 years; how do they make such claims?

- *Accelerated life test*: Parts under stress (higher load, higher temperature, etc.) will usually fail sooner than parts that are unstressed. By modeling the lifetimes of parts under various stresses, we can estimate (extrapolate to) the lifetime of parts that are unstressed.

- Example: resin glue failure time

Example — Resin Glue Failure Time[1]

- Goal: to estimate the life time (in hours) of an encapsulating resin for gold-aluminum bonds in integrated circuits (operating at 120°C)
- Method: accelerated life test
- Design: Randomly assign 37 units to one of 5 different temperature stresses (in Celsius)

$$175^\circ, 194^\circ, 213^\circ, 231^\circ, 250^\circ$$

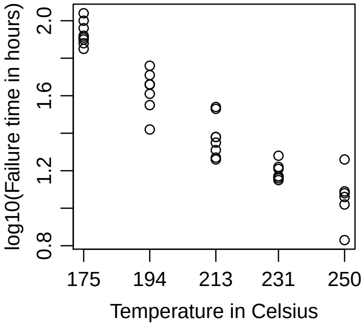
- Treatments: temperature in Celsius
- Response: $Y = \log_{10}$(time to failure in hours) of the tested material.

[1] Source: p. 448-449, *Accelerated Testing* (Nelson 2004). Original data is provided by Dr. Muhib Khan of AMD.

Example — Resin Glue Failure Time — Data

$$Y = \log_{10}(\text{Failure time in hours})$$

Temperature (°C)				
175	194	213	231	250
2.04	1.66	1.53	1.15	1.26
1.91	1.71	1.54	1.22	0.83
2.00	1.42	1.38	1.17	1.08
1.92	1.76	1.31	1.16	1.02
1.85	1.66	1.35	1.21	1.09
1.96	1.61	1.27	1.28	1.06
1.88	1.55	1.26	1.17	
1.90	1.66	1.38		



Data file: resin.txt

Example — Resin Glue Failure Time — SS_{trt}

Temperature (°C)	175	194	213	231	250
Size n_i	8	8	8	7	6
Mean $\overline{y}_{i\bullet}$	1.9325	1.62875	1.3775	1.1943	1.0567
SD s_i	0.0634	0.1048	0.1071	0.0458	0.1384

$$
\begin{aligned}
\overline{y}_{\bullet\bullet} &= \frac{\sum n_i \overline{y}_{i\bullet}}{N} \\
&= \frac{1}{37}(8 \cdot 1.9325 + 8 \cdot 1.62875 + 8 \cdot 1.3775 + 7 \cdot 1.1943 + 6 \cdot 1.0567) \\
&\approx 1.4651
\end{aligned}
$$

$$
\begin{aligned}
SS_{trt} &= \sum_{i=1}^{g} \sum_{j=1}^{n_i} (\overline{y}_{i\bullet} - \overline{y}_{\bullet\bullet})^2 = \sum_{i=1}^{5} n_i (\overline{y}_{i\bullet} - \overline{y}_{\bullet\bullet})^2 \\
&= 8(1.9325 - 1.4651)^2 + 8(1.62875 - 1.4651)^2 + 8(1.3775 - 1.4651)^2 \\
&\quad + 7(1.1943 - 1.4651)^2 + 6(1.0567 - 1.4651)^2 \\
&\approx 3.543
\end{aligned}
$$

Example: Resin Glue Failure Time — SSE, F, and P-value

Temperature (°C)	175	194	213	231	250
Size n_i	8	8	8	7	6
Mean $\overline{y}_{i\bullet}$	1.9325	1.62875	1.3775	1.1943	1.0567
SD s_i	0.0634	0.1048	0.1071	0.0458	0.1384

$$
\begin{aligned}
SSE &= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij} - \overline{y}_{i\bullet})^2 = \sum_{i=1}^{g}(n_i - 1)s_i^2 \\
&= (8-1)(0.0634)^2 + (8-1)(0.1048)^2 + (8-1)(0.1071)^2 \\
&\quad + (7-1)(0.0458)^2 + (6-1)(0.1384)^2 \\
&\approx 0.2937
\end{aligned}
$$

Example: Resin Glue Failure Time — SSE, F, and P-value

Temperature (°C)	175	194	213	231	250
Size n_i	8	8	8	7	6
Mean $\overline{y}_{i\bullet}$	1.9325	1.62875	1.3775	1.1943	1.0567
SD s_i	0.0634	0.1048	0.1071	0.0458	0.1384

$$\begin{aligned}
SSE &= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij}-\overline{y}_{i\bullet})^2 = \sum_{i=1}^{g}(n_i-1)s_i^2 \\
&= (8-1)(0.0634)^2 + (8-1)(0.1048)^2 + (8-1)(0.1071)^2 \\
&\quad + (7-1)(0.0458)^2 + (6-1)(0.1384)^2 \\
&\approx 0.2937
\end{aligned}$$

$$F\text{-statistic} = \frac{SS_{trt}/(g-1)}{SSE/(N-g)} = \frac{3.543/(5-1)}{0.2937/(37-5)} \approx 96.52$$

with $g-1=5-1=4$ and $N-g=37-5=32$ degrees of freedom.

Example: Resin Glue Failure Time — SSE, F, and P-value

Temperature (°C)	175	194	213	231	250
Size n_i	8	8	8	7	6
Mean $\overline{y}_{i\bullet}$	1.9325	1.62875	1.3775	1.1943	1.0567
SD s_i	0.0634	0.1048	0.1071	0.0458	0.1384

$$\begin{aligned} SSE &= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij} - \overline{y}_{i\bullet})^2 = \sum_{i=1}^{g}(n_i - 1)s_i^2 \\ &= (8-1)(0.0634)^2 + (8-1)(0.1048)^2 + (8-1)(0.1071)^2 \\ &\quad + (7-1)(0.0458)^2 + (6-1)(0.1384)^2 \\ &\approx 0.2937 \end{aligned}$$

$$F\text{-statistic} = \frac{SS_{trt}/(g-1)}{SSE/(N-g)} = \frac{3.543/(5-1)}{0.2937/(37-5)} \approx 96.52$$

with $g - 1 = 5 - 1 = 4$ and $N - g = 37 - 5 = 32$ degrees of freedom.

The P-value is $\approx 2.189 \times 10^{-17}$. The data exhibit strong evidence against the H_0 that all means are equal.

```
> pf(96.52, df1 = 4, df2 = 32, lower.tail=F)
[1] 2.188913e-17
```

Always Check the Degrees of Freedom!

‘

```
> resin = read.table(
     "http://www.stat.uchicago.edu/~yibi/s222/resin.txt", h=T)
> str(resin)
'data.frame': 37 obs. of  2 variables:
 $ tempC: int  175 175 175 175 175 175 175 175 194 194 ...
 $ y    : num  2.04 1.91 2 1.92 1.85 1.96 1.88 1.9 1.66 1.71 ...
> lm1 = lm(y ~ tempC, data=resin)
> anova(lm1)
          Df Sum Sq Mean Sq F value    Pr(>F)
tempC      1 3.4593  3.4593  325.41 < 2.2e-16 ***
Residuals 35 0.3721  0.0106
```

Something wrong?

Always Check the Degrees of Freedom!

‘

```
> resin = read.table(
     "http://www.stat.uchicago.edu/~yibi/s222/resin.txt", h=T)
> str(resin)
'data.frame': 37 obs. of  2 variables:
 $ tempC: int  175 175 175 175 175 175 175 175 194 194 ...
 $ y    : num  2.04 1.91 2 1.92 1.85 1.96 1.88 1.9 1.66 1.71 ...
> lm1 = lm(y ~ tempC, data=resin)
> anova(lm1)
          Df Sum Sq Mean Sq F value    Pr(>F)
tempC      1 3.4593  3.4593  325.41 < 2.2e-16 ***
Residuals 35 0.3721  0.0106
```

Something wrong?

d.f. for `tempC` should be $g - 1 = 5 - 1 = 4$, not 1.

Always Check the Degrees of Freedom!

‘

```
> resin = read.table(
     "http://www.stat.uchicago.edu/~yibi/s222/resin.txt", h=T)
> str(resin)
'data.frame': 37 obs. of  2 variables:
 $ tempC: int  175 175 175 175 175 175 175 175 194 194 ...
 $ y    : num  2.04 1.91 2 1.92 1.85 1.96 1.88 1.9 1.66 1.71 ...
> lm1 = lm(y ~ tempC, data=resin)
> anova(lm1)
          Df Sum Sq Mean Sq F value    Pr(>F)
tempC      1 3.4593  3.4593  325.41 < 2.2e-16 ***
Residuals 35 0.3721  0.0106
```

Something wrong?

d.f. for `tempC` should be $g - 1 = 5 - 1 = 4$, not 1. As `tempC` is *numerical*, by default, R will fit the regression model

$$y_{ij} = \beta_0 + \beta_1 \texttt{tempC}_i + \varepsilon_{ij}.$$

The ANOVA table above is for comparing the regression model above with the null model $y_{ij} = \beta_0 + \varepsilon_{ij}$.

Always Check the Degrees of Freedom

To fit the multi-sample model in Lecture 1, which the textbook called the means model

$$y_{ij} = \mu_i + \varepsilon_{ij}.$$

we need to let R treat `tempC` as *categorical* by `as.factor()`ing it.

```
> lmmeans = lm(y ~ as.factor(tempC), data=resin)
> anova(lmmeans)
                 Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(tempC)  4 3.5376 0.88441  96.363 < 2.2e-16 ***
Residuals        32 0.2937 0.00918
```

Always Check the Degrees of Freedom

To fit the multi-sample model in Lecture 1, which the textbook called the means model

$$y_{ij} = \mu_i + \varepsilon_{ij}.$$

we need to let R treat `tempC` as *categorical* by `as.factor()`ing it.

```
> lmmeans = lm(y ~ as.factor(tempC), data=resin)
> anova(lmmeans)
                 Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(tempC)  4 3.5376 0.88441  96.363 < 2.2e-16 ***
Residuals        32 0.2937 0.00918
```

What's the difference between the regression model and the means model?

Means Model Is a Multiple Linear Regression Model

For an experiment with g treatments, the Means model

$$y_{ij} = \mu_i + \varepsilon_{ij}$$

can be written as a multiple linear regression model by defining a *dummy variable* for each treatment group. The dummy variable for the ith treatment is defined as

$$D_i = \begin{cases} 1 & \text{if the experimental unit receives the } i\text{th treatment} \\ 0 & \text{otherwise} \end{cases}$$

The means model can then be written as a regression model

$$y = \mu_1 D_1 + \mu_2 D_2 + \cdots + \mu_g D_g + \varepsilon$$

Means Model Is a Multiple Linear Regression Model

For an experiment with g treatments, the Means model

$$y_{ij} = \mu_i + \varepsilon_{ij}$$

can be written as a multiple linear regression model by defining a *dummy variable* for each treatment group. The dummy variable for the ith treatment is defined as

$$D_i = \begin{cases} 1 & \text{if the experimental unit receives the } i\text{th treatment} \\ 0 & \text{otherwise} \end{cases}$$

The means model can then be written as a regression model

$$y = \mu_1 D_1 + \mu_2 D_2 + \cdots + \mu_g D_g + \varepsilon$$

- If a unit receives the 2nd treatment, then $D_2 = 1$ and $D_i = 0$ for $i \neq 2$, then

$$y = \mu_1 \cdot 0 + \mu_2 \cdot 1 + \mu_3 \cdot 0 + \cdots + \mu_g \cdot 0 + \varepsilon = \mu_2 + \varepsilon$$

Means Model Is a Multiple Linear Regression Model

For an experiment with g treatments, the Means model

$$y_{ij} = \mu_i + \varepsilon_{ij}$$

can be written as a multiple linear regression model by defining a *dummy variable* for each treatment group. The dummy variable for the ith treatment is defined as

$$D_i = \begin{cases} 1 & \text{if the experimental unit receives the } i\text{th treatment} \\ 0 & \text{otherwise} \end{cases}$$

The means model can then be written as a regression model

$$y = \mu_1 D_1 + \mu_2 D_2 + \cdots + \mu_g D_g + \varepsilon$$

- If a unit receives the 2nd treatment, then $D_2 = 1$ and $D_i = 0$ for $i \neq 2$, then

$$y = \mu_1 \cdot 0 + \mu_2 \cdot 1 + \mu_3 \cdot 0 + \cdots + \mu_g \cdot 0 + \varepsilon = \mu_2 + \varepsilon$$

- This regression model has *no intercept*

In R, putting −1 in the model formula tells R to fit a regression model with no intercept.

```
> lmmeans = lm(y ~ -1 + as.factor(tempC), data = resin)
> summary(lmmeans)
                    Estimate Std. Error t value Pr(>|t|)
as.factor(tempC)175  1.93250    0.03387   57.05   <2e-16 ***
as.factor(tempC)194  1.62875    0.03387   48.09   <2e-16 ***
as.factor(tempC)213  1.37750    0.03387   40.67   <2e-16 ***
as.factor(tempC)231  1.19429    0.03621   32.98   <2e-16 ***
as.factor(tempC)250  1.05667    0.03911   27.02   <2e-16 ***
```

In R, putting −1 in the model formula tells R to fit a regression model with no intercept.

```
> lmmeans = lm(y ~ -1 + as.factor(tempC), data = resin)
> summary(lmmeans)
                    Estimate Std. Error t value Pr(>|t|)
as.factor(tempC)175  1.93250    0.03387   57.05   <2e-16 ***
as.factor(tempC)194  1.62875    0.03387   48.09   <2e-16 ***
as.factor(tempC)213  1.37750    0.03387   40.67   <2e-16 ***
as.factor(tempC)231  1.19429    0.03621   32.98   <2e-16 ***
as.factor(tempC)250  1.05667    0.03911   27.02   <2e-16 ***
```

Temp	n_i	$\bar{y}_{i\bullet}$	
175	8	1.93250	
194	8	1.62875	
213	8	1.37750	
231	7	1.19429	
250	6	1.05667	

In R, putting −1 in the model formula tells R to fit a regression model with no intercept.

```
> lmmeans = lm(y ~ -1 + as.factor(tempC), data = resin)
> summary(lmmeans)
                    Estimate Std. Error t value Pr(>|t|)
as.factor(tempC)175  1.93250    0.03387   57.05   <2e-16 ***
as.factor(tempC)194  1.62875    0.03387   48.09   <2e-16 ***
as.factor(tempC)213  1.37750    0.03387   40.67   <2e-16 ***
as.factor(tempC)231  1.19429    0.03621   32.98   <2e-16 ***
as.factor(tempC)250  1.05667    0.03911   27.02   <2e-16 ***
```

Temp	n_i	$\overline{y}_{i\bullet}$	$\mathrm{SE}(\overline{y}_{i\bullet}) = \sqrt{\mathrm{MSE}/n_i}$
175	8	1.93250	$\sqrt{0.00918/8} = 0.03387$
194	8	1.62875	$\sqrt{0.00918/8} = 0.03387$
213	8	1.37750	$\sqrt{0.00918/8} = 0.03387$
231	7	1.19429	$\sqrt{0.00918/7} = 0.03621$
250	6	1.05667	$\sqrt{0.00918/6} = 0.03911$

In R, putting -1 in the model formula tells R to fit a regression model with no intercept.

```
> lmmeans = lm(y ~ -1 + as.factor(tempC), data = resin)
> summary(lmmeans)
                    Estimate Std. Error t value Pr(>|t|)
as.factor(tempC)175  1.93250    0.03387   57.05   <2e-16 ***
as.factor(tempC)194  1.62875    0.03387   48.09   <2e-16 ***
as.factor(tempC)213  1.37750    0.03387   40.67   <2e-16 ***
as.factor(tempC)231  1.19429    0.03621   32.98   <2e-16 ***
as.factor(tempC)250  1.05667    0.03911   27.02   <2e-16 ***
```

Temp	n_i	$\overline{y}_{i\bullet}$	$\text{SE}(\overline{y}_{i\bullet}) = \sqrt{\text{MSE}/n_i}$
175	8	1.93250	$\sqrt{0.00918/8} = 0.03387$
194	8	1.62875	$\sqrt{0.00918/8} = 0.03387$
213	8	1.37750	$\sqrt{0.00918/8} = 0.03387$
231	7	1.19429	$\sqrt{0.00918/7} = 0.03621$
250	6	1.05667	$\sqrt{0.00918/6} = 0.03911$

Observe the `Estimate` coefficients are simply group means $\overline{y}_{i\bullet}$ and `Std. Error` is the $\text{SE}(\overline{y}_{i\bullet}) = \sqrt{\text{MSE}/n_i}$, for a group mean μ_i introduced in Lecture 2, where $\text{MSE} = 0.00918$ for the resin data.

Means Model and the Effects Model

The textbook models for multi-sample data in two forms:

$$
\begin{aligned}
y_{ij} &= \mu_i + \varepsilon_{ij} && \text{(means model)} \\
&= \mu + \alpha_i + \varepsilon_{ij} && \text{(effects model)}
\end{aligned}
$$

- Observe the effects model has $g + 1$ parameters $\mu, \alpha_1, \ldots, \alpha_g$, while the means model only has g parameters $\mu_1, \ldots, \mu_g$

Means Model and the Effects Model

The textbook models for multi-sample data in two forms:

$$\begin{aligned} y_{ij} &= \mu_i + \varepsilon_{ij} && \text{(means model)} \\ &= \mu + \alpha_i + \varepsilon_{ij} && \text{(effects model)} \end{aligned}$$

- Observe the effects model has $g + 1$ parameters $\mu, \alpha_1, \ldots, \alpha_g$, while the means model only has g parameters $\mu_1, \ldots, \mu_g$
- The effects model is **overparameterized**, meaning it has more parameters than required. One can change the values of μ and α_i's as follows without changing the value of $\mu + \alpha_i$.

$$\begin{array}{ccc} \mu & \to & \mu + c \\ \alpha_1 & \to & \alpha_1 - c \\ \vdots & & \vdots \\ \alpha_g & \to & \alpha_g - c \end{array}$$

Thus parameters in the effects model **cannot be uniquely determined**.

Means Model and the Effects Model

The textbook models for multi-sample data in two forms:

$$\begin{aligned} y_{ij} &= \mu_i + \varepsilon_{ij} && \text{(means model)} \\ &= \mu + \alpha_i + \varepsilon_{ij} && \text{(effects model)} \end{aligned}$$

- Observe the effects model has $g + 1$ parameters $\mu, \alpha_1, \ldots, \alpha_g$, while the means model only has g parameters $\mu_1, \ldots, \mu_g$
- The effects model is **overparameterized**, meaning it has more parameters than required. One can change the values of μ and α_i's as follows without changing the value of $\mu + \alpha_i$.

$$\begin{array}{ccc} \mu & \to & \mu + c \\ \alpha_1 & \to & \alpha_1 - c \\ \vdots & & \vdots \\ \alpha_g & \to & \alpha_g - c \end{array}$$

  Thus parameters in the effects model **cannot be uniquely determined**.
- The two models are equivalent in the sense that they give identical fitted values

How to Deal With Overparametrization?

A common way to deal with overparametrization is forcing, α_1, or one of the α_i's, to be 0. Then

$$\mathbb{E}[y_{ij}] = \begin{cases} \mu_1 = \mu & \text{for trt 1} \\ \mu_2 = \mu + \alpha_2 & \text{for trt 2} \\ \vdots & \vdots \\ \mu_g = \mu + \alpha_g & \text{for trt g} \end{cases} \Rightarrow \begin{aligned} \mu &= \mu_1 \\ \alpha_i &= \mu_i - \mu_1 \\ &\text{for } i = 2, 3, \ldots, g \end{aligned}$$

- Testing $\alpha_i = 0$ is equivalent to testing $\mu_i = \mu_1$
  Useful for comparing treatments

How to Deal With Overparametrization?

A common way to deal with overparametrization is forcing, α_1, or one of the α_i's, to be 0. Then

$$\mathbb{E}[y_{ij}] = \begin{cases} \mu_1 = \mu & \text{for trt 1} \\ \mu_2 = \mu + \alpha_2 & \text{for trt 2} \\ \vdots & \vdots \\ \mu_g = \mu + \alpha_g & \text{for trt g} \end{cases} \Rightarrow \begin{array}{rl} \mu &= \mu_1 \\ \alpha_i &= \mu_i - \mu_1 \\ & \text{for } i = 2, 3, \ldots, g \end{array}$$

- Testing $\alpha_i = 0$ is equivalent to testing $\mu_i = \mu_1$
  Useful for comparing treatments

Another way to cope with overparametrization is forcing α_i's add up to 0

$$\sum_{i=1}^{g} \alpha_i = 0$$

This **sum-to-zero constraint** seems to come up abruptly, but it can greatly simplify the formulas for factorial design models in Chapter 8. We will come back to it in Chapter 8.

Effects Model

```
> lmeffects = lm(y ~ as.factor(tempC), data = resin)
> summary(lmeffects)
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.93250    0.03387  57.055  < 2e-16 ***
as.factor(tempC)194 -0.30375    0.04790  -6.341 4.06e-07 ***
as.factor(tempC)213 -0.55500    0.04790 -11.586 5.49e-13 ***
as.factor(tempC)231 -0.73821    0.04958 -14.889 6.13e-16 ***
as.factor(tempC)250 -0.87583    0.05174 -16.928  < 2e-16 ***
```

Note there is no `as.factor(temp)175` since R sets $\alpha_{175} = 0$.

Temp	n_i	$\overline{y}_{i\bullet}$	$\widehat{\alpha}_i = \overline{y}_{i\bullet} - \overline{y}_{1\bullet}$
175	8	1.933	$1.933 - 1.933 = 0$
194	8	1.629	$1.629 - 1.933 = -0.304$
213	8	1.378	$1.378 - 1.933 = -0.555$
231	7	1.194	$1.194 - 1.933 = -0.737$
250	6	1.057	$1.057 - 1.933 = -0.876$

Effects Model

```
> lmeffects = lm(y ~ as.factor(tempC), data = resin)
> summary(lmeffects)
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.93250    0.03387  57.055  < 2e-16 ***
as.factor(tempC)194 -0.30375    0.04790  -6.341 4.06e-07 ***
as.factor(tempC)213 -0.55500    0.04790 -11.586 5.49e-13 ***
as.factor(tempC)231 -0.73821    0.04958 -14.889 6.13e-16 ***
as.factor(tempC)250 -0.87583    0.05174 -16.928  < 2e-16 ***
```

Note there is no `as.factor(temp)175` since R sets $\alpha_{175} = 0$.

Temp	n_i	$\overline{y}_{i\bullet}$	$\widehat{\alpha}_i = \overline{y}_{i\bullet} - \overline{y}_{1\bullet}$	$\mathrm{SE}(\overline{y}_{i\bullet} - \overline{y}_{1\bullet}) = \sqrt{\mathrm{MSE}(\frac{1}{n_i} + \frac{1}{n_1})}$
175	8	1.933	$1.933 - 1.933 = 0$	0
194	8	1.629	$1.629 - 1.933 = -0.304$	$\sqrt{0.00918(\frac{1}{8} + \frac{1}{8})} = 0.04790$
213	8	1.378	$1.378 - 1.933 = -0.555$	$\sqrt{0.00918(\frac{1}{8} + \frac{1}{8})} = 0.04790$
231	7	1.194	$1.194 - 1.933 = -0.737$	$\sqrt{0.00918(\frac{1}{7} + \frac{1}{8})} = 0.04958$
250	6	1.057	$1.057 - 1.933 = -0.876$	$\sqrt{0.00918(\frac{1}{6} + \frac{1}{8})} = 0.05174$

Comparison of Two ANOVA Tables

For comparing the means models $y_{ij} = \mu_i + \varepsilon_{ij}$ against the null models $y_{ij} = \mu + \varepsilon_{ij}$:

Source	df	SS	MS	F
Treatments	$g-1$	$\text{SS}_{trt} = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\bar{y}_{i\bullet} - \bar{y}_{\bullet\bullet})^2$	$\text{MS}_{trt} = \dfrac{\text{SS}_{trt}}{g-1}$	$\dfrac{\text{MS}_{trt}}{\text{MSE}}$
Error	$N-g$	$\text{SSE} = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_{i\bullet})^2$	$\text{MSE} = \dfrac{\text{SSE}}{N-g}$	
Total	$N-1$	$\text{SST} = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_{\bullet\bullet})^2$		

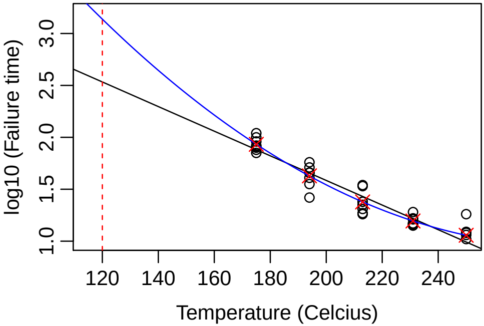
For comparing the MLR models $y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \varepsilon_i$ against the null model $y_i = \beta_0 + \varepsilon_i$:

Source	df	SS	MS	F
Regression	p	$\text{SSR} = \sum_{i=1}^{n}(\widehat{y}_i - \bar{y})^2$	$\text{MSR} = \dfrac{\text{SSR}}{p}$	$F = \dfrac{\text{MSR}}{\text{MSE}}$
Error	$n-p-1$	$\text{SSE} = \sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$	$\text{MSE} = \dfrac{\text{SSE}}{n-p-1}$	
Total	$n-1$	$\text{SST} = \sum_{i=1}^{n}(y_i - \bar{y})^2$		

Limitation of ANOVA F-Tests

The tiny P-value of ANOVA F-test merely shows resin glue at different temperatures has different lifetimes.

```
                 Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(tempC)  4 3.5376 0.88441  96.363 < 2.2e-16 ***
Residuals        32 0.2937 0.00918
```



(red cross marks group means)

However, our ultimate goal is to predict the lifetime of the glue at temperature 120°C.

Dose-Response Modeling

In some experiments, the treatments are associated with *numerical levels* x_i such as drug dose, baking time, or temperature.

Textbook refers to such levels as *doses*.

Dose-Response Modeling

In some experiments, the treatments are associated with *numerical levels* x_i such as drug dose, baking time, or temperature.

Textbook refers to such levels as *doses*.

- The means model $y_{ij} = \mu_i + \varepsilon_{ij}$ **doesn't specify how the response** y **changes with the treatment levels** x_i. Hence it cannot predict y at dose level$x$ not observed in the experiment

Dose-Response Modeling

In some experiments, the treatments are associated with *numerical levels* x_i such as drug dose, baking time, or temperature.

Textbook refers to such levels as *doses*.

- The means model $y_{ij} = \mu_i + \varepsilon_{ij}$ **doesn't specify how the response** y **changes with the treatment levels** x_i. Hence it cannot predict y at dose level$x$ not observed in the experiment
- With a *numerical* treatment factor, researchers are usually more interested on how the response is affected as a function of the dose level x_i

$$y_{ij} = f(x_i; \theta) + \varepsilon_{ij},$$

e.g.,

$$\begin{aligned} f(x_i; \beta_0, \beta_1) &= \beta_0 + \beta_1 x_i; \\ f(x_i; \beta_0, \beta_1, \beta_2) &= \beta_0 + \beta_1 x_i + \beta_2 x_i^2; \text{ or} \\ f(x_i; \beta_0, \beta_1) &= \beta_0 + \beta_1 \log(x_i). \end{aligned}$$

$$y_{ij} = f(x_i; \theta) + \varepsilon_{ij}$$

Advantages of dose-response modeling

- less complex (fewer parameters)
- easier to interpret (sometimes)
- can predict y at dose levels not observed in the experiment

$$y_{ij} = f(x_i;\theta) + \varepsilon_{ij}$$

Advantages of dose-response modeling

- less complex (fewer parameters)
- easier to interpret (sometimes)
- can predict y at dose levels not observed in the experiment

Issues to consider:

- How to choose the function f?

$$y_{ij} = f(x_i; \theta) + \varepsilon_{ij}$$

Advantages of dose-response modeling

- less complex (fewer parameters)
- easier to interpret (sometimes)
- can predict y at dose levels not observed in the experiment

Issues to consider:

- How to choose the function f?
    - One commonly used family of functions f are *polynomials*:

$$f(x_i; \beta) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k,$$

$$y_{ij} = f(x_i; \theta) + \varepsilon_{ij}$$

Advantages of dose-response modeling

- less complex (fewer parameters)
- easier to interpret (sometimes)
- can predict y at dose levels not observed in the experiment

Issues to consider:

- How to choose the function f?
  - One commonly used family of functions f are *polynomials*:

$$f(x_i; \beta) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k,$$

Polynomials are NOT always the best choice.

$$y_{ij} = f(x_i; \theta) + \varepsilon_{ij}$$

Advantages of dose-response modeling

- less complex (fewer parameters)
- easier to interpret (sometimes)
- can predict y at dose levels not observed in the experiment

Issues to consider:

- How to choose the function f?
  - One commonly used family of functions f are *polynomials*:

    $$f(x_i; \beta) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k,$$

    Polynomials are NOT always the best choice.
- How to assess how well f fits the data? ...... Goodness of fit

Polynomial Models

Let t_i denote the temperature in treatment group i.
Consider the following polynomial models for the resin glue data.

$$\begin{aligned}
\text{Null} &: y_{ij} = \mu + \varepsilon_{ij} \\
\text{Linear} &: y_{ij} = \beta_0 + \beta_1 t_i + \varepsilon_{ij} \\
\text{2nd order} &: y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \varepsilon_{ij} \\
\text{3rd order} &: y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \varepsilon_{ij} \\
\text{4th order} &: y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \beta_4 t_i^4 + \varepsilon_{ij}
\end{aligned}$$

- For simplicity, we would choose the <u>lowest</u> possible order of polynomial that adequately fits the data.

Polynomial Models

Let t_i denote the temperature in treatment group i.
Consider the following polynomial models for the resin glue data.

$$\begin{aligned}
\text{Null} &: y_{ij} = \mu + \varepsilon_{ij} \\
\text{Linear} &: y_{ij} = \beta_0 + \beta_1 t_i + \varepsilon_{ij} \\
\text{2nd order} &: y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \varepsilon_{ij} \\
\text{3rd order} &: y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \varepsilon_{ij} \\
\text{4th order} &: y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \beta_4 t_i^4 + \varepsilon_{ij}
\end{aligned}$$

- For simplicity, we would choose the <u>lowest</u> possible order of polynomial that adequately fits the data.
- Every model is nested in the model below it. (Why?)

Polynomial Models

Let t_i denote the temperature in treatment group i.
Consider the following polynomial models for the resin glue data.

$$\begin{aligned}
\text{Null}: & \; y_{ij} = \mu + \varepsilon_{ij} \\
\text{Linear}: & \; y_{ij} = \beta_0 + \beta_1 t_i + \varepsilon_{ij} \\
\text{2nd order}: & \; y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \varepsilon_{ij} \\
\text{3rd order}: & \; y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \varepsilon_{ij} \\
\text{4th order}: & \; y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \beta_4 t_i^4 + \varepsilon_{ij}
\end{aligned}$$

- For simplicity, we would choose the <u>lowest</u> possible order of polynomial that adequately fits the data.
- Every model is nested in the model below it. (Why?)
- Never skip a term. If a higher order term is significant, e.g., t_i^3, than all lower order terms have to be kept $(1, t_i, t_i^2)$, even if they are not significant.

Polynomial Models

Let t_i denote the temperature in treatment group i.
Consider the following polynomial models for the resin glue data.

$$\begin{aligned}
\text{Null} &: y_{ij} = \mu + \varepsilon_{ij} \\
\text{Linear} &: y_{ij} = \beta_0 + \beta_1 t_i + \varepsilon_{ij} \\
\text{2nd order} &: y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \varepsilon_{ij} \\
\text{3rd order} &: y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \varepsilon_{ij} \\
\text{4th order} &: y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \beta_4 t_i^4 + \varepsilon_{ij}
\end{aligned}$$

- For simplicity, we would choose the lowest possible order of polynomial that adequately fits the data.
- Every model is nested in the model below it. (Why?)
- Never skip a term. If a higher order term is significant, e.g., t_i^3, than all lower order terms have to be kept (1, t_i, t_i^2), even if they are not significant.
- Why not consider 5th order or higher order models?

In general, for an experiment with g treatment groups, if the treatment factor is numeric, one can fit a polynomial model up to degree $g-1$

$$y_{ij} = \beta_0 + \beta_1 x_i + \cdots + \beta_{g-1} x_i^{g-1} + \varepsilon_{ij}.$$

Question: For the resin glue data, what will happen if a 5th-order polynomial model is fitted?

$$y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \beta_4 t_i^4 + \beta_5 t_i^5 + \varepsilon_{ij}$$

In general, for an experiment with g treatment groups, if the treatment factor is numeric, one can fit a polynomial model up to degree $g-1$

$$y_{ij} = \beta_0 + \beta_1 x_i + \cdots + \beta_{g-1} x_i^{g-1} + \varepsilon_{ij}.$$

Question: For the resin glue data, what will happen if a 5th-order polynomial model is fitted?

$$y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \beta_4 t_i^4 + \beta_5 t_i^5 + \varepsilon_{ij}$$

- There are 6 parameters $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$, but only 5 groups — overparametrization!

In general, for an experiment with g treatment groups, if the treatment factor is numeric, one can fit a polynomial model up to degree $g-1$

$$y_{ij} = \beta_0 + \beta_1 x_i + \cdots + \beta_{g-1} x_i^{g-1} + \varepsilon_{ij}.$$

Question: For the resin glue data, what will happen if a 5th-order polynomial model is fitted?

$$y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \beta_4 t_i^4 + \beta_5 t_i^5 + \varepsilon_{ij}$$

- There are 6 parameters $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$, but only 5 groups — overparametrization!
- There exist more than one 5th-order polynomial passing through the 5 points $(175, \mu_1)$, $(194, \mu_2)$, $(213, \mu_3)$, $(231, \mu_4)$, and $(250, \mu_5)$. Thus the 6 coefficients $\beta_0, \beta_1, \ldots, \beta_5$ CANNOT be uniquely determined.

In general, for an experiment with g treatment groups, if the treatment factor is numeric, one can fit a polynomial model up to degree $g-1$

$$y_{ij} = \beta_0 + \beta_1 x_i + \cdots + \beta_{g-1} x_i^{g-1} + \varepsilon_{ij}.$$

Question: For the resin glue data, what will happen if a 5th-order polynomial model is fitted?

$$y_{ij} = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \beta_4 t_i^4 + \beta_5 t_i^5 + \varepsilon_{ij}$$

- There are 6 parameters $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$, but only 5 groups — overparametrization!
- There exist more than one 5th-order polynomial passing through the 5 points $(175, \mu_1)$, $(194, \mu_2)$, $(213, \mu_3)$, $(231, \mu_4)$, and $(250, \mu_5)$. Thus the 6 coefficients $\beta_0, \beta_1, \ldots, \beta_5$ CANNOT be uniquely determined.

A rule of thumb: for an experiment with g treatments, we can fit a model with at most g parameters.

Linear Model (1)

Let's try fitting the linear model: $y_{ij} = \beta_0 + \beta_1 t_i + \varepsilon_{ij}$.

```
> lm1 = lm(y ~ tempC, data = resin)
> summary(lm1)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.9560075  0.1391174   28.44   <2e-16 ***
tempC       -0.0118567  0.0006573  -18.04   <2e-16 ***
```

- Fitted equation: $\log_{10}(\text{failure time}) = 3.956 - 0.01186\,T$
- Predicted $\log_{10}(\text{failure time})$ at 120° is

$$3.956 - 0.01186 \times 120 \approx 2.5332,$$

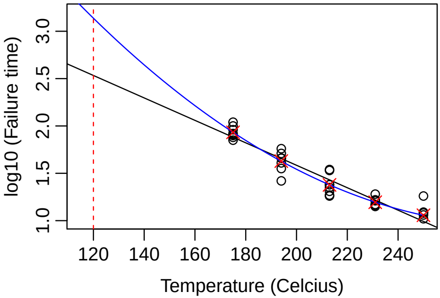
and hence the failure time at 120° is predicted as

$$10^{2.5332} \approx 341 \text{ hours.}$$

Linear Model (2)

R commands for the predicted log10(failure time) along with a 95% prediction interval:

```
> predict(lm1, newdata=data.frame(tempC=120), interval="prediction")
       fit      lwr      upr
1 2.533201 2.289392 2.777011
```



By imposing the regression line on the top of the scatter plot, we can see y is a slightly curved with temperature. Using the linear model, the failure time at 120° will be *underestimated*.

2nd Order Model

```
> lm2 = lm(y ~ tempC+I(tempC^2), data=resin)
> summary(lm2)
(... part of the output is omitted ...)
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.4179987  1.1564331   6.415 2.51e-07 ***
tempC        -0.0450981  0.0110542  -4.080 0.000258 ***
I((tempC)^2)  0.0000786  0.0000261   3.011 0.004879 **
```

- Fitted model: $\log_{10}(\text{time}) = 7.418 - 0.0451\,T + 0.0000786\,T^2$
- Predicted log10(time) at 120° is

$$7.418 - 0.0451 \times 120 + 0.0000786 \times (120)^2 \approx 3.138$$

The predicted failure time at 120° is $10^{3.138} \approx 1374$ hours.

3rd and 4th Order Models

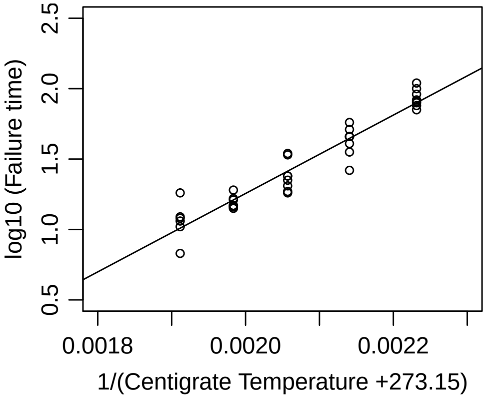
```
> lm3 = lm(y ~ tempC+I(tempC^2)+I(tempC^3), data = resin)
> summary(lm3)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.827e+00  1.299e+01   0.526    0.603
tempC       -3.659e-02  1.865e-01  -0.196    0.846
I(tempC^2)   3.815e-05  8.860e-04   0.043    0.966
I(tempC^3)   6.357e-08  1.392e-06   0.046    0.964

> lm4 = lm(y ~ tempC+I(tempC^2)+I(tempC^3)+I(tempC^4), data = resin)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.699e-01  1.957e+02   0.005    0.996
tempC        7.573e-02  3.750e+00   0.020    0.984
I(tempC^2)  -7.649e-04  2.679e-02  -0.029    0.977
I(tempC^3)   2.600e-06  8.459e-05   0.031    0.976
I(tempC^4)  -2.988e-09  9.962e-08  -0.030    0.976
```

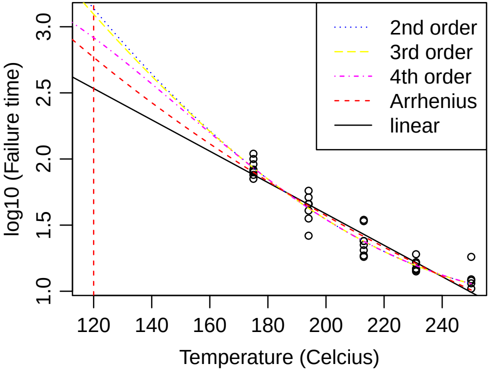
Arrhenius Law

The Arrhenius rate law in Thermodynamics says, *the log of failure time is linear in the inverse of absolute Kelvin temperature, which equals the Centigrade temperature plus 273.15 degrees.*

$$\text{Arrhenius Model: } y_{ij} = \beta_0 + \frac{\beta_1}{T + 273.15}.$$



```
> lmarr = lm(y ~ I(1/(tempC+273.15)), data=resin)
> summary(lmarr)
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)               -4.3120     0.3007  -14.34  3.2e-16 ***
I(1/(tempC + 273.15))   2783.7764   144.6808   19.24  < 2e-16 ***
```



Predicted $\log_{10}$(failure time) at 120° is $-4.312 + \frac{2783.78}{120+273.15} \approx 2.77$.
The predicted failure time is $e^{2.77} \approx 588$ hours.

Data Can Distinguish Models Only at Observed Dose Levels

In addition to polynomial models and the Arrhenius model, many other models can be considered

$$
\begin{aligned}
y_{ij} &= \beta_0 + \beta_1 \log(t_i) + \varepsilon_{ij}, \\
y_{ij} &= \beta_0 + \beta_1 \exp(t_i) + \varepsilon_{ij}, \\
y_{ij} &= \beta_0 + \beta_1 \sin(t_i) + \varepsilon_{ij}, \\
y_{ij} &= \beta_0 + f(t_i) + \varepsilon_{ij}.
\end{aligned}
$$

Data Can Distinguish Models Only at Observed Dose Levels

In addition to polynomial models and the Arrhenius model, many other models can be considered

$$\begin{aligned} y_{ij} &= \beta_0 + \beta_1 \log(t_i) + \varepsilon_{ij}, \\ y_{ij} &= \beta_0 + \beta_1 \exp(t_i) + \varepsilon_{ij}, \\ y_{ij} &= \beta_0 + \beta_1 \sin(t_i) + \varepsilon_{ij}, \\ y_{ij} &= \beta_0 + f(t_i) + \varepsilon_{ij}. \end{aligned}$$

As we only have observations at 5 temperatures:

$$175,\ 194,\ 213,\ 231,\ 250,$$

the data cannot distinguish between two models:

$$y_{ij} = f(t_i) + \varepsilon_{ij} \quad \text{and} \quad y_{ij} = g(t_i) + \varepsilon_{ij},$$

if $f(t)$ and $g(t)$ coincide at $t = 175, 194, 213, 231, 250$, even if f and g behave differently in other places.

The Model that Fits the Data the Best

If no restriction is placed on f, how well the model $y_{ij} = f(t_i) + \varepsilon_{ij}$ can possibly fit the data?

The Model that Fits the Data the Best

If no restriction is placed on f, how well the model $y_{ij} = f(t_i) + \varepsilon_{ij}$ can possibly fit the data?

The least square method will choose the f that minimize

$$\sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij}-f(t_i))^2 = \sum_{j=1}^{n_1}(y_{1j}-f(t_1))^2+\cdots+\sum_{j=1}^{n_g}(y_{gj}-f(t_g))^2$$

The Model that Fits the Data the Best

If no restriction is placed on f, how well the model $y_{ij} = f(t_i) + \varepsilon_{ij}$ can possibly fit the data?

The least square method will choose the f that minimize

$$\sum_{i=1}^{g} \sum_{j=1}^{n_i} (y_{ij} - f(t_i))^2 = \sum_{j=1}^{n_1} (y_{1j} - f(t_1))^2 + \cdots + \sum_{j=1}^{n_g} (y_{gj} - f(t_g))^2$$

Recall that given a list of numbers $x_1, x_2, \ldots, x_n$ the c that minimize $\sum_{i=1}^{n} (x_i - c)^2$ is the mean $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$.

The Model that Fits the Data the Best

If no restriction is placed on f, how well the model $y_{ij} = f(t_i) + \varepsilon_{ij}$ can possibly fit the data?

The least square method will choose the f that minimize

$$\sum_{i=1}^{g} \sum_{j=1}^{n_i} (y_{ij} - f(t_i))^2 = \sum_{j=1}^{n_1} (y_{1j} - f(t_1))^2 + \cdots + \sum_{j=1}^{n_g} (y_{gj} - f(t_g))^2$$

Recall that given a list of numbers $x_1, x_2, \ldots, x_n$ the c that minimize $\sum_{i=1}^{n} (x_i - c)^2$ is the mean $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$.

Thus the least square method will choose the f that

$$f(t_i) = \overline{y}_{i\bullet}.$$

The Model that Fits the Data the Best

If no restriction is placed on f, how well the model $y_{ij} = f(t_i) + \varepsilon_{ij}$ can possibly fit the data?

The least square method will choose the f that minimize

$$\sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij}-f(t_i))^2 = \sum_{j=1}^{n_1}(y_{1j}-f(t_1))^2+\cdots+\sum_{j=1}^{n_g}(y_{gj}-f(t_g))^2$$

Recall that given a list of numbers $x_1, x_2, \ldots, x_n$ the c that minimize $\sum_{i=1}^{n}(x_i - c)^2$ is the mean $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Thus the least square method will choose the f that

$$f(t_i) = \overline{y}_{i\bullet}.$$

Thus the smallest SSE a model $y_{ij} = f(t_i) + \varepsilon_{ij}$ can possibly achieve is

$$\sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij} - \overline{y}_{i\bullet})^2$$

which is the SSE for the **means model** $y_{ij} = \mu_i + \varepsilon_{ij}$.

The Model that Fits the Data the Best

If no restriction is placed on f, how well the model $y_{ij} = f(t_i) + \varepsilon_{ij}$ can possibly fit the data?

The least square method will choose the f that minimize

$$\sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij}-f(t_i))^2 = \sum_{j=1}^{n_1}(y_{1j}-f(t_1))^2+\cdots+\sum_{j=1}^{n_g}(y_{gj}-f(t_g))^2$$

Recall that given a list of numbers $x_1, x_2, \ldots, x_n$ the c that minimize $\sum_{i=1}^{n}(x_i - c)^2$ is the mean $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Thus the least square method will choose the f that

$$f(t_i) = \overline{y}_{i\bullet}.$$

Thus the smallest SSE a model $y_{ij} = f(t_i) + \varepsilon_{ij}$ can possibly achieve is

$$\sum_{i=1}^{g}\sum_{j=1}^{n_i}(y_{ij} - \overline{y}_{i\bullet})^2$$

which is the SSE for the **means model** $y_{ij} = \mu_i + \varepsilon_{ij}$.

Conclusion: **no other models can beat the means model in minimizing the SSE**.

Goodness of Fit

As the means model is the model that fit the data the best, we can access the goodness of a model $y_{ij} = f(t_i) + \varepsilon_{ij}$ by comparing it with the means model.

$$\text{Full Model} : y_{ij} = \mu_i + \varepsilon_{ij}$$
$$\text{Reduced Model} : y_{ij} = f(t_i) + \varepsilon_{ij}$$

This comparison is legitimate because any model $y_{ij} = f(t_i) + \varepsilon_{ij}$ is nested in the means model $y_{ij} = \mu_i + \varepsilon_{ij}$ (letting $\mu_i = f(t_i)$).

We can use the F-statistic below for comparing a reduced model and a full model

$$F = \frac{(SSE_{reduced} - SSE_{full})/(df_{reduced} - df_{full})}{SSE_{full}/df_{full}}$$

If we get a small P-value, H0 is rejected, which means that the reduced model doens't fit as good as the means model.
If we get a large P-value, fail to reject H0, then it means the reduced model fit the data nearly as good as the best (means model).

Goodness of Fit of the Linear Model

Since the linear model (reduced model) is nested in the means model (full), use the F-statistic for model comparison we get

```
> lm1 = lm(y ~ tempC, data = resin)                # linear model
> lmmeans = lm(y ~ as.factor(tempC), data = resin) # means model
> anova(lm1,lmmeans)
Analysis of Variance Table

Model 1: y ~ tempC
Model 2: y ~ as.factor(tempC)
  Res.Df     RSS Df Sum of Sq      F  Pr(>F)
1     35 0.37206
2     32 0.29369  3   0.07837 2.8463 0.05303 .
```

The P-value 0.05303 is moderate evidence showing the linear doesn't fit the data so well.

Goodness of Fit of the 2nd-Order Model

Since the 2nd-order model (reduced model) is also nested in the means model (full model), again using the F-statistic for model comparison we get

```
> lm2 = lm(y ~ tempC+I((tempC)^2), data=resin)      # 2nd-order model
> lmmeans = lm(y ~ as.factor(tempC), data = resin)  # means model
> anova(lm2,lmmeans)
Analysis of Variance Table

Model 1: y ~ tempC + I((tempC)^2)
Model 2: y ~ as.factor(tempC)
  Res.Df     RSS Df  Sum of Sq      F Pr(>F)
1     34 0.29372
2     32 0.29369  2 2.6829e-05 0.0015 0.9985
```

The large p-value 0.9985 shows the 2nd-order model fits the data nearly as good as the best model. Does this indicate the 2nd-order model is an adequate model?

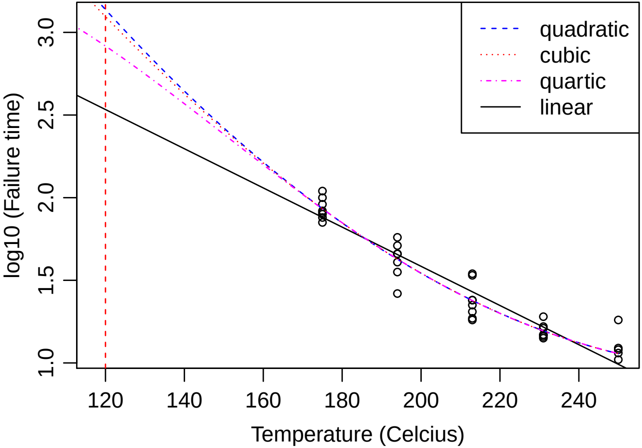
Shall We Consider a 3rd- or 4th-Order Model?

No. Because

$$\text{2nd-order} \subset \text{3rd-order} \subset \text{4th-order} \subset \text{Means Model}$$

the 3rd- or 4th-order model won't fit the data better than the means model does. As the 2nd-order model fits the data nearly as well as the means model, the 4 models just fit as well as each other. In this case we simply choose the model of lowest complexity.

Be Cautious About Extrapolation



Though the 2nd-, 3rd-, 4th-order model fit the 5 points nearly as well, their predicted values at 120°C are quite different,

$$\text{2nd-order} > \text{3rd-order} > \text{4th-order} > \text{linear}$$

Since the Arrhenius model is nested in the means model, we can check its goodness of fit.

```
> lmarr = lm(y ~ I(1/(tempC+273.15)), data=resin)   # Arrhenius model
> lmmeans = lm(y ~ as.factor(tempC), data = resin)  # means model
> anova(lmarr, lmmeans)
Analysis of Variance Table

Model 1: y ~ I(1/(tempC + 273.15))
Model 2: y ~ as.factor(tempC)
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1     35 0.33093
2     32 0.29369  3  0.037239 1.3525 0.2749
```

The moderately large P-value 0.2749 told us the Arrhenius Model is acceptable relative to the best model.